1996-05

The Multiple Vehicle TSP with Time Windows and Equity Constraints over a Multiple Day Horizon

Dell, Robert F.

Institute for Operations Research and the Management Sciences

Transportation Science Vol. 30, No. 2 (May 1996); p. 120-133.
http://hdl.handle.net/10945/48881

The Multiple Vehicle TSP with Time Windows and Equity Constraints over a Multiple Day Horizon[1]

ROBERT F. DELL

Operations Research Department, Naval Postgraduate School, Monterey, California 93943-5000

RAJAN BATTA and MARK H. KARWAN

Department of Industrial Engineering, State University of New York at Buffalo, Buffalo, New York 14260

This paper considers the multiple vehicle traveling salesman problem cast over a multiple day routing scenario, with time windows and equity constraints. It develops an optimal column generation procedure and heuristic variations which solve test problems derived from the overnight delivery business. Extensive empirical testing indicates the judicious chose of a few routes for each day can incorporate equity at little or no additional cost.

The overnight delivery business, though relatively new, draws fierce competition. Customers have many overnight carriers to choose from (*e.g.*, Federal Express, UPS, DHL), and hence factors such as convenience (*e.g.*, how far is the closest drop box; how efficient is the pickup service), reliability (*e.g.*, is the package delivered within the time-frame promised), and price become important to retain business. Motivating this paper is one aspect of customer satisfaction—maintaining equity in delivery times. The importance of maintaining equity in delivery times stems from the need to preserve business from large-volume customers. A large-volume customer wants his/her package delivered not just on time but "first thing in the morning". In other words, they expect better delivery than promised. This is easy to do, if the number of large-volume customers is relatively small, by assigning one such customer to each route and placing him/her first on the route. However, the solution is not clear when the number of such customers substantially exceeds the number of delivery vehicles available. If one set of routes is repeatedly used, some large-volume customers could receive significantly better service than others. Any large-volume customer who consistently does not receive his/her packages "first thing in the morning" could seek an alternative carrier. One "solution" to this problem is a multiple day routing scenario which allows a different sets of routes on each day. Such a modeling framework leads to a multiple vehicle traveling salesman problem cast over a multiple day scenario, with time windows and equity constraints.

Incorporating equity in operations research models is not new. Previous examples include papers by MANDELL (1991) who presents a general overview of modeling equity in public systems; by GOPALAN ET AL. (1990) who model equity in the transportation of hazardous materials; by LARSON (1987) who cites equity as a factor in people's perception of justice in a queuing system; and by KEENEY (1980a,b) who expresses equity as the magnitude of the largest differences in the level of risk among a fixed set of individuals.

This paper introduces the problem of routing vehicles subject to an equity constraint. It provides a mathematical formulation of the problem, introduces both optimal and heuristic solution methods, and performs an extensive computational study. The following sections present: 1) the formulation; 2) an

[1]Accepted by Mark S. Daskin.

0041-1655/96/3002-0120 $01.25

overview of our solution procedure; 3) the test problems; 4) empirical results; and 5) a summary.

1. *MM*-TSPTW FORMULATION

THIS PAPER PRESENTS a variation of the vehicle routing problems described in BODIN ET AL. (1983), GOLDEN and ASSAD (1987), GOLDEN and ASSAD (1988), LAWLER ET AL. (1985), and MAGNANTI (1981). The difference is both the inclusion of equity constraints between customers' average delivery times and the use of a multiple day horizon. We refer to the problem as the *modified* multiple traveling salesman problem with time windows (*m*m-TSPTW).

1.1. Column Generation Formulation

Column generation has been used successfully by a number of authors to optimally solve routing problems (see *e.g.*, AGARWAL, MATHUR, and SALKIN, 1989; DESROSIERS, SOUMIS, and DESROCHERS, 1984; and DESROCHERS, DESROSIERS, and SOLOMON, 1990). An extension of DESROSIERS ET AL. provides the framework for the *m*m-TSPTW master and subproblem formulations, as detailed below.

1.1.1. Master Problem

The master problem formulation uses the following notation:

- Indices:

i ≡node ($i = 0, 1, \ldots, n, n+1$), where 0 represents the depot from which routes commence, $n+1$ is the ending depot, and 1 through n correspond to customers requiring delivery on each day.

r ≡route that provides feasible delivery to a subset of the customers.

d ≡delivery day.

g ≡group.

- Data

Ω ≡set of routes.

S_g ≡set of customers in group g.

δ_g ≡equity parameter for group g which specifies one-half the maximum permissible deviation in average delivery time.

σ_{ri} = 1 if route r delivers to customer i and 0 otherwise.

τ_{ri} ≡time route r delivers to customer i (considered 0 when route r does not deliver to customer i).

D ≡number of days over which equity is sought.

c_r ≡cost of using route r (includes both the fixed cost of vehicle usage and variable routing costs).

- Decision Variables:

y_{rd} = 1 if route r is taken on day d and 0 otherwise.

t^g ≡a measure of delivery time for group g. This variable represents the range midpoint for customer delivery in the group.

- The master problem formulation, constrained set partition ($\mathscr{CSP}$), is as follows:

$$\text{minimize } Z_{\mathscr{CSP}} = \sum_{r\in\Omega} c_r\left(\sum_{d=1}^{D} y_{rd}\right)$$

subject to the constraints:

$$\sum_{r\in\Omega} \sigma_{ri}\, y_{rd} = 1 \qquad i = 1, 2, \ldots, n \qquad d = 1, 2, \ldots, D \tag{1}$$

$$\sum_{r\in\Omega}\sum_{d=1}^{D} \tau_{ri} y_{rd} - t^g \le D\delta_g \quad i \in S_g \ \forall g \tag{2}$$

$$t^g - \sum_{r\in\Omega}\sum_{d=1}^{D} \tau_{ri} y_{rd} \le D\delta_g \quad i \in S_g \ \forall g \tag{2'}$$

$$y_{rd} \text{ binary } r \in \Omega, \qquad d = 1, 2, \ldots, D \tag{3}$$

Constraint set (1) dictates that each customer must be serviced by exactly one vehicle on each day. Set partitioning constraints are necessary since set covering constraints (used by previous column generation techniques) can result in multiple deliveries to a customer which artificially satisfy the equity constraints. Constraint sets (2) and (2′) ensure that customers within the same priority group receive approximately equal levels of service. (They can alternately be combined into a single constraint with the addition of a bounded variable.)

1.1.2. Subproblem for Day *d*

The subproblem identifies routes with favorable reduced cost to the linear programming (*lp*) relaxation of $\mathscr{CSP}$ when Ω contains a subset of all possible routes. Using the optimal dual multipliers to the *lp* relaxation, the subproblem seeks the route for day d with the minimum $\mathscr{CSP}$ reduced cost. When the subproblem finds a route with favorable reduced cost, it is added to the set Ω, otherwise the subproblem verifies the *lp* relaxation to $\mathscr{CSP}$ as optimal. The subproblem for day d uses the following additional notation:

- Indices:

j, k ≡aliases used interchangeably with i (previously defined).

- Data:

$[e_i, l_i]$ ≡ time window for customer i. Specifically, e_i (l_i) represents the earliest (latest) allowable delivery time for customer i.
T_{ij} ≡ travel time between customers i and j.
c_{ij} ≡ cost of travel between customers i and j.
M_{ij} ≡ constant to insure appropriate delivery times for routes ($M_{ij} \geq l_i + T_{ij} - e_i$).
w_{id} ≡ optimal dual multiplier associated with constraint (1) in the lp relaxation of $\mathscr{CSP}$. This dual multiplier provides a measure of the cost associated with having to deliver to customer i on day d.
$u_i(u_i')$ ≡ optimal dual multiplier associated with constraint (2) (2′) in the lp relaxation of $\mathscr{CSP}$. Taken together, these multipliers indicate a relative desire to deliver early or late within a customer's time window.

- Decision Variables:

x_{ij} ≡ 1 if customer i immediately precedes customer j and 0 otherwise.
t_i ≡ delivery time for customer i.

The route, r, for day d with the most favorable reduced cost is mathematically stated as:

$$\text{minimize: } c_r - \sum_{i=1}^{n} w_{id}\sigma_{ri} + \sum_{i=1}^{n}(u_i - u_i')\tau_{ri}$$

subject to the condition that σ_{ri}, τ_{ri}, c_r form a feasible route.

By the definition of x_{ij}, $c_r = \sum_{i=0}^{n}\sum_{j=1}^{n+1} c_{ij}x_{ij}$ and σ_{ri} equals one if and only if $\sum_{j=1}^{n+1} x_{ij} = 1$. Thus, $(c_r - \sum_{i=1}^{n} w_{id}\sigma_{ri})$ can be viewed as $\sum_{i=0}^{n}\sum_{j=1}^{n+1}(c_{ij} - w_{id})x_{ij}$, with the understanding that $w_{0d} = 0$. The $\sum_{i=1}^{n}(u_i - u_i')\tau_{ri}$ term translates into the expression $\sum_{i=1}^{n}(u_i - u_i')t_i$ using the delivery time variable t_i (*i.e.*, the value of t_i becomes the data τ_{ri} in the master problem). This provides the shortest path with time ($\mathscr{SPT}$) window formulation:

$$\text{minimize } Z_{\mathscr{SPT}} = \sum_{i=1}^{n}(u_i - u_i')t_i + \sum_{i=0}^{n}\sum_{j=1}^{n+1}(c_{ij} - w_{id})x_{ij}$$

subject to the constraints:

$$\sum_{i=0}^{n} x_{ik} = \sum_{j=1}^{n+1} x_{kj} \qquad k = 1, 2, \ldots, n \tag{4}$$

$$\sum_{i=0}^{n} x_{i0} = \sum_{j=1}^{n+1} x_{0j} = 1 \tag{5}$$

$$e_i \leq t_i \leq l_i \qquad i = 0, 1, \ldots, n, n+1 \tag{6}$$

$$t_i + T_{ij} - t_j \leq M_{ij}(1 - x_{ij}) \qquad i,j = 0, 1, \ldots, n, n+1 \tag{7}$$

$$x_{ij} \text{ binary } i, j = 0, 1, \ldots, n, n+1$$

Constraint set (4) ensures conservation of flow. Constraint set (5) enforces the leaving and entering of a depot. Constraint set (6) ensures customer delivery within the appropriate time window. Constraint set (7) provides consistency between delivery times and the developed path as well as the elimination of cycles. $\mathscr{SPT}$ is a shortest path problem with time windows (SPPTW) which possesses an additional cost on the nodal arrival time. The next section presents the characteristics of the paths generated by $\mathscr{SPT}$ which restrict the set Ω to be finite, and demonstrates that integer solutions can be formed from a fractional lp relaxation of $\mathscr{CSP}$.

1.2. Properties of Ω

Consider a delivery sequence for m customers, [1], [2], . . . , [m], delivered by the same vehicle, where [i] denotes the i^{th} customer in the sequence. Let the delivery time for the first j customers in the sequence be fixed at times $t_{[1]}, t_{[2]}, \ldots, t_{[j]}$, respectively. Then the conditional earliest and latest customer delivery times $Ca_{[i]}$ and $Cb_{[i]}$ for the remaining $m - j$ customers (if feasible) is recursively given by:

$$Ca_{[i]} = \max[e_{[i]}, Ca_{[i-1]} + T_{[i-1],[i]}],$$
$$\text{for } j+1 \leq i \leq m, \quad \text{where} \quad Ca_{[j]} = t_{[j]}; \text{ and} \tag{8}$$

$$Cb_{[i]} = \min[l_{[i]}, Cb_{[i+1]} - T_{[i],[i+1]}],$$
$$\text{for } j+1 \leq i \leq m-1, \quad \text{where} \quad Cb_{[m]} = l_{[m]}. \tag{9}$$

We define a *window limited route* for the delivery sequence [1], [2], . . . , [m] as a feasible sequence where $t_{[i]}$, the delivery time for the i^{th} customer in the sequence, is either $Ca_{[i]}$ or $Cb_{[i]}$, when these quantities are calculated recursively starting with $j = 1$ and proceeding to $j = m - 1$. Specifically, $t_{[1]}$ can be either $Ca_{[1]}$ or $Cb_{[1]}$. Then, $t_{[2]}$ can be either $Ca_{[2]}$ or $Cb_{[2]}$ (when feasible), as given by (8) and (9) respectively and so on.

Property 1 verifies that the subproblem only generates window limited routes and Property 2 shows feasible delivery routes may be represented as a convex combination of window limited routes.

PROPERTY 1. *Given ties are broken by choosing the lowest possible delivery time, the optimal solution to $\mathscr{SPT}$ must be a window limited route.*

The result is shown by contradiction assuming a route containing m nodes is generated that is not a window limited route. Given delivery time (t_i) and corresponding costs $(u_i - u_i')$, the route's time of delivery contributes cost $\sum_{i=1}^{m}(u_i - u_i')t_i$. Since this route is not a window limited route, there exists a $[j]$ such that $Ca_{[j]} < t_{[j]} < Cb_{[j]}$. This implies either:

1. $(u_{[j]} - u'_{[j]})Ca_{[j]} < (u_{[j]} - u'_{[j]})t_{[j]}$.
   A contradiction since $\mathcal{SPT}$ is optimally solved and therefore $t_{[j]}$ would have been set to $Ca_{[j]}$.
2. $(u_{[j]} - u'_{[j]})Ca_{[j]} > (u_{[j]} - u'_{[j]})t_{[j]}$.
   A contradiction since the procedure would have obtained a lower reduced cost by setting $t_{[j]}$ as late as possible at value $Cb_{[j]}$. □

PROPERTY 2. *A feasible delivery route may be represented as a convex combination of window limited routes.*

Let $t_{[1]}, t_{[2]}, \ldots, t_{[m]}$ be the delivery times for the m customers on a feasible delivery route. Let j be such that $t_{[1]}, t_{[2]}, \ldots, t_{[j-1]}$ correspond to delivery times on a window limited route w for customers [1], [2], . . . , $[j - 1]$, and $t_{[j]}$ has value other than $Ca_{[j]}$ and $Cb_{[j]}$. Clearly $j \geq 1$. Define two extensions of the window limited route w to include customer $[j]$ with delivery times $Ca_{[j]}$ and $Cb_{[j]}$ labeled w^1 and w^2 respectively. Clearly, $Ca_{[j]} < t_{[j]} < Cb_{[j]}$ and hence the portion of the feasible delivery route with customer $[j]$ can be viewed as a convex combination of window limited routes w^1 and w^2. It follows that any feasible delivery route can be viewed as a convex combinations of window limited routes. □

These properties follow directly from Dantzig-Wolfe decomposition (DANTZIG and WOLFE (1960) and (1961)) since window limited routes are extreme points to $\mathcal{CSP}$. They highlight that the *lp* relaxation to $\mathcal{CSP}$ may contain fractional variables which combine to form feasible integer solutions. Furthermore, a convex combination of window limited routes may be the only feasible integer solution, as demonstrated by a simple two customer example where a separate vehicle delivers to each customer, any travel time is possible, exactly equal delivery times must be obtained over 1 day, $[e_1, l_1] = [1, 3]$, and $[e_2, l_2] = [2, 4]$. Any equal delivery time for both customers in the interval [2, 3] is feasible. However, routes generated by the subproblem would only deliver customer 1 at times 1 or 3 and customer 2 at times 2 or 4. A convex combination of window limited routes would be needed to obtain a feasible integer solution.

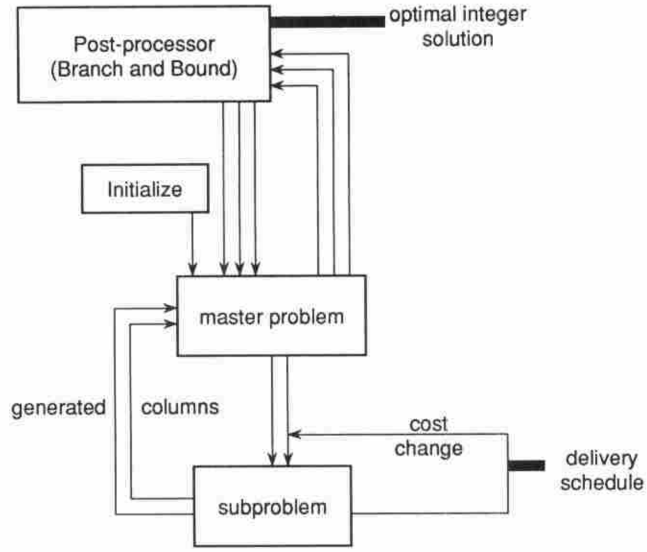

Fig. 1. Overview of column generation procedure.

2. SOLUTION PROCEDURE

FIGURE 1 OVERVIEWS the solution procedure and shows its hierarchical nature using a different number of arcs to link components. For example, through sufficient iterations, the single arc exiting the subproblem box must be satisfied before continuing to the double arched loop. In the same manner, through a sufficient number of iterations, the double arched loop must be satisfied before progressing to the tripled arched, post-processor, loop.

2.1. Initialization

Our column generation procedure starts with a feasible solution to $\mathcal{CSP}$ constructed by using one vehicle for every customer on each day. This artificial solution allows any feasible delivery time for each customer and therefore the easy construction of a feasible solution. The *lp* relaxation of the master problem then supplies the dual multipliers as parameters to the subproblem.

2.2. $\mathcal{SPT}$ Heuristic (subproblem solution)

DESROSIERS ET AL. (1984) use cyclic shortest paths within their master problem. Unlike their situation, a cyclic solution cannot be allowed in our procedure due to the equity constraints and the potential for them to be artificially satisfied by multiple deliveries on a single day. DELL (1990) presents an optimal algorithm to solve the subproblem using Lagrangian relaxation to remove cyclic solutions. Since this optimal procedure is computationally intensive, it is replaced by a heuristic (the $\mathcal{SPT}$ heuristic) empirically found to be an effective substitute (see DELL (1990)).

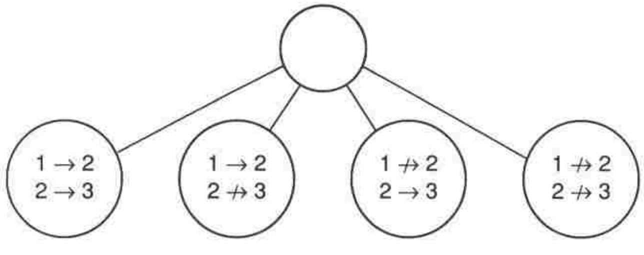

Fig. 2. Branching scheme.

The $\mathcal{SPT}$ heuristic employs a permanent node labeling procedure similar to DESROCHERS and SOUMIS (1988), with multiple labels at each node. Since the dual multipliers associated with constraints (1), (2), and (2′) dictate the cost of arriving at any node is a linear function of time, all node labels have linear cost functions. Additionally, any label also contains a valid time interval and path employed to obtain the corresponding cost. Keeping track of the path forming a label, allows us to prohibit cyclic solutions. It is this non-cyclic limitation which causes the procedure to be heuristic.

2.3. Post-Processor

Some apparently fractional solutions may correspond to integer solutions since convex combinations of window limited paths may form feasible integer solutions. The post-processor checks each linear programming solution for such occurrences.

The post-processor also attempts to improve the *lp* bound by the addition of a cutting plane which imposes an integral number of vehicles. As in Desrosiers et al., this is a valid inequality since the fixed cost of any route exceeds any potential variable routing cost (see Section 3). With this cost structure, it becomes obvious that the number of vehicles must be the ceiling of the number obtained by the *lp* relaxation.

Usually the *lp* relaxation to $\mathcal{CSP}$ is not integer and a branch-and-bound (b&b) scheme modified from Desrosiers et al. is used. The scheme obtains the best solution by selectively enforcing partial sequences and allowing the corresponding delivery times to be decided by the master problem. The subproblem enforces partial sequences by the elimination of arcs. As an example, assume route $1 \rightarrow 2 \rightarrow 3$ is fractional in the *lp* relaxation to $\mathcal{CSP}$. Figure 2 shows the four reduced problems where $i \rightarrow j$ implies customer i immediately precedes customer j on any route containing i (accomplished in the subproblem by eliminating all arcs $i \rightarrow k$ with $k \neq j$) and $i \nrightarrow j$ implies that j never directly follows i (eliminate arc $i \rightarrow j$). When branching, routes violating these conditions are removed from the master problem and strictly enforced in all routes generated by the subproblem.

Within the above type of branching, $\mathcal{CSP}$ maintains the same structure and the routes are not fixed. Only the sequence is fixed. This allows the corresponding delivery time to be determined within the master problem by picking a window limited route or a convex combination of window limited routes. This is an important feature of the branching scheme since fixing delivery times could restrict routes that might exist in the optimal solution.

The main difficulty with this form of branching is that a fractional route containing m customers results in $\mathbb{O}(2^m)$ reduced problems to solve. Due to this exponential nature, heuristic branching rules are used, as described below with other techniques empirically found to speed solution time.

2.4. Delivery Schedule

We define a delivery schedule as a sufficient number of routes to provide delivery to all customers. As shown in Figure 1, a delivery schedule may be generated at each call to the subproblem. Furthermore, a delivery schedule provides a feasible solution to the delivery problem in the absence of equity considerations.

The procedure to generate a delivery schedule is straightforward. Remove customers from the initial route generated by the subproblem from consideration and resolve the subproblem to obtain a new route. Repeat this process until one generates a set of routes that taken together cover all customers. This constitutes a delivery schedule (which may or may not satisfy equity constraints).

To form a delivery schedule the subproblem initially uses costs from the *lp* relaxation to $\mathcal{CSP}$. After generating a route with unfavorable reduced cost, all costs change to values such that:

- $u_i' = 0$ for all i;
- $u_i > u_j(1 + \delta_{\max})$ where customer i is higher priority than customer j and $\delta_{\max}$ is the maximum difference in delivery time between two customers; and
- $(c_{ij} - w_{id}) = -\max_k\{u_k t_{\max}\}$ for all i, j pairs where $t_{\max}$ is the maximum possible delivery time for any customer.

These costs ensure it is always more beneficial to include an additional customer on a route and it is preferable to have customer i before customer j if both exist in the same sequence.

2.5. Heuristic Variations

The following heuristics have been shown empirically to speed the computations without significantly sacrificing solution quality.

2.5.1. Multiple Day Heuristic

Computational requirements dramatically grow when considering multiple days (the number of possible routing combinations increases exponentially in the number of days). To overcome the computational burden, a heuristic solves the multiple day problem as a series of single day problems. This heuristic uses the solution from a one day problem with restrictions (2) and (2′) non-binding (this solution should closely correspond to the minimum operating cost). Keeping in mind the delivery times for the customers on this first day (by adding appropriate constants to constraints (2) and (2′)), the heuristic solves for routes on the second day. When obtaining the second day's routes, the equity for the two day problem is strictly enforced. Of course, if it is desired to obtain equity in more than two days, a series of one day problems can be recursively solved while appropriately adjusting the equity constraints.

2.5.2. Heuristic Fathom

Empirical work demonstrates that a large percentage of fathomed nodes are eliminated by bounding. This presents both good and bad news. On the bright side, it implies that imposed equity constraints do not significantly limit routing possibilities (*i.e.*, it gives the reassurance that equity may be enforced at little additional cost). On the other hand, the number of possible routing sequences is not significantly limited by the equity constraints. Therefore, the quality of bound and speed with which one can be found is important to increase fathoming. We use a depth first search of the b&b tree.

To increase possible fathoming by bounding, we heuristically fathomed nodes within fixed percentages of the incumbent solution and found the number of nodes requiring analysis could be more than halved by fathoming any node on the b&b tree within 2% to 5% of the incumbent solution. All results in Section 4 come from fathoming any solution within 2% of the incumbent.

3. TEST PROBLEMS

FEW BENCHMARK DATA sets exist for vehicle routing and scheduling problems with time windows. In a recent paper, KOSKOSIDIS, POWELL, and SOLOMON (1992) use what they consider the only standardized problems, originally from SOLOMON (1984). Unfortunately, Solomon's data sets are unsuitable for our purposes since he randomly derives time windows of many varying lengths whereas time windows in our application have wide intervals of only a few possible durations. Due to this difference, we develop data sets from information supplied by Federal Express Corporation.

Data sets come from information on 1,448 deliveries representing one day's service in the Palo Alto, California, region. This delivery region, located between latitudes 37.308689 to 37.538776 and longitudes 122.097527 to 122.514534, represents approximately 425 square miles. Figure 3 shows an overview of the region and location of customers.

Using the spatial locations and additional delivery information, an extension of the work by KOLESAR, WALKER, and HAUSNER (1975) provides the following model:

$$T(L) = \begin{cases} 3.2, & \text{if } L \le 0.05, \\ 3.2 + 3.3\sqrt{L}, & \text{if } 0.05 < L \le 7.0, \\ 3.2 + 4.4 + 0.6L, & \text{if } L > 7.0 \end{cases}$$

where $T(L)$ represents the minutes of travel and service time needed to cover a distance of L miles. Conceptually, this model includes a fixed time associated with any delivery of 3.2 minutes, assumes the vehicle has a variable velocity for distances less than 7.0 miles, and a constant velocity for distances above 7.0 mile. With this travel time model, the travel time between every two customers is calculated.

Our computational study uses data sets comprised of 25, 35, and 50 aggregate customers. On average, each aggregate customer represents four customers. The travel times are adjusted to represent the expected travel and service time for an aggregate group. We report 3 instances for each size problem and refer to a specific problem by the number of aggregate customers with the instance in parentheses. For example, 25(1), 25(2), and 25(3) respectively denote the first, second, and third data set of 25 aggregate customers. In general, instance (1) has aggregate customers relatively spread out, instance (2) has aggregate customers relatively clustered, and instance (3) the aggregate customers relatively semi-clustered (*i.e.*, groups of customers clustered together).

All problem instances have two priority groups, $\mathscr{HP}$ (high priority) and $\mathscr{LP}$ (low priority) where approximately one third are high priority. Since enforcing equity for customers is potentially costly, one would not consider too many customers to be labeled high priority and we consider one third to be an upper bound. With only a few high priority customers, a solution, under any equity condition, could

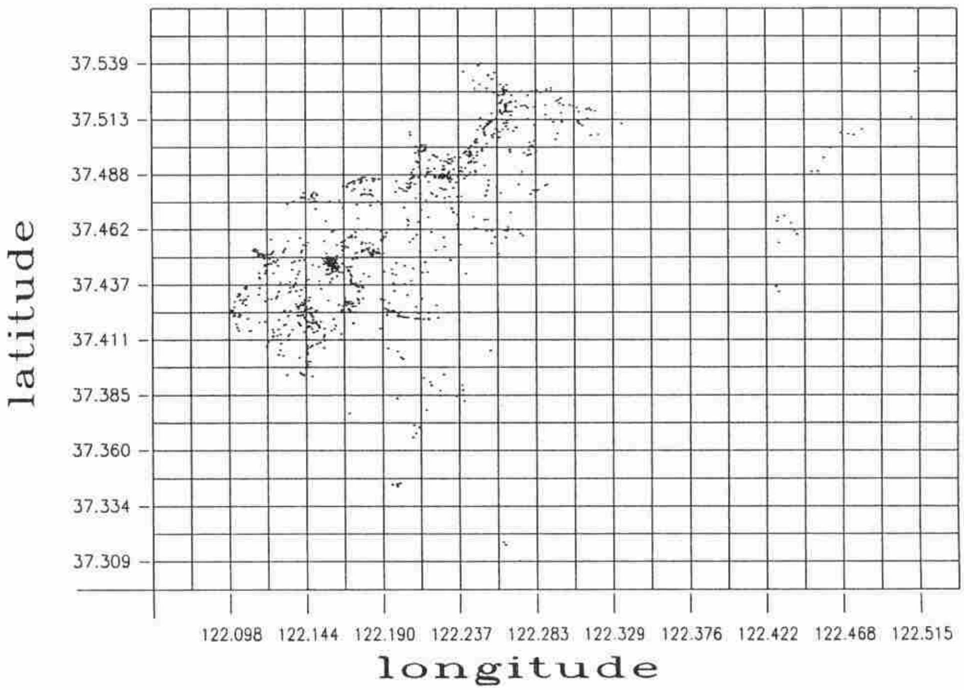

Fig. 3. Spatial overview of provided data.

easily be found by placing each customer first on each route. Since too few high priority customers provides little or no algorithmic challenge, we conduct our study on the upper bound.

Low priority customers have a time window of [7:30,10:30]. For high priority customers, a time window of [7:30,9:00] accounts for their status. Of course, the time window limits for high priority customers can have a large effect on the computational results. For our empirical study, 9:00 a.m. is an upper bound which does not significantly limit possible routes. Although greater restriction would undoubtedly improve our algorithmic performance, the results may not be representative of all cases.

In addition to the data above, the estimated fixed cost of vehicle use is \$148 and a vehicle's variable cost is \$0.136 per minute.

4. EMPIRICAL RESULTS

ALL ALGORITHMS ARE coded in C primarily to allow the dynamic nature of our column generation, shortest path, and b&b to be easily handled without excessive and unnecessary memory requirements. All CPU statistics come from a Sun 4/490.

We solve each test problem under three different levels of equity for $\mathscr{HP}$ customers with the equity for $\mathscr{LP}$ customers kept at an extremely loose level. Considering the fact that deliveries for $\mathscr{HP}$ customers may be taken from [7:30, 9:00], or one and one-half hours, $\delta_{\mathscr{HP}} = 30$ minutes is a loose level of equity which would place some restriction on possible solutions. Using this as a base, $\delta_{\mathscr{HP}} = 20$ minutes and $\delta_{\mathscr{HP}} = 10$ minutes are moderate and tight levels of imposed equity.

Table I contains a representative sample of results. A column-by-column description follows:

- (Column 1) Equity level in minutes, where the first number corresponds to $\delta_{\mathscr{HP}}$. Recall that $\delta_{\mathscr{HP}}$ is our equity parameter and $2\delta_{\mathscr{HP}}$ represents the maximum allowed (per day) difference in delivery time for customers in group $\mathscr{HP}$. The number in parentheses is the number of days and the letter h signifies the heuristic use for the multiple day scenario.
- (Grouped Column 2) Heuristic Solutions. This section of columns presents solutions generated by repeatedly solving our shortest path problem (depicted in Figure 1 as the delivery schedule). Since generating delivery schedules requires extra and sometimes unnecessary computation time, this portion of our algorithm is only applied before entering the b&b procedure. The table shows the three heuristic solutions that provide the:

—lowest cost (given in a per day basis).

—lowest equity violation on average from the average delivery time of customers of group $\mathscr{HP}$, referred to as equity violation (1). Shown

TABLE I

Empirical Results for 25(1)

	Heuristic Solutions			First Int. Soln.		Best Integer Solution				CPU Required			Branch & Bound			
		Equ. viol.						% deviation								
Equity Level	Obj. val.	(1)	(2)	Obj. val.	CPU (secs)	Obj. val.	CPU (secs)	cut	w/o cut	Total	% simp.	% gen. col.	No. expl.	No. fath. (bound)	No. fath. (infea.)	No. act.
30(1)	702	0.0(0)	0.0	852	10.3	702	78.1	0.0	0.0	100.4	31	57	—	—	—	—
	702	0.0(0)	0.0													
	702	0.0(0)	0.0													
20(1)	851	20.7(3)	16.0	852	303.4	852	303.4	0.1	21.0	303.5	53	38	14	109	4	—
	852	6.5(6)	16.0													
	702	19.3(4)	23.0													
20(2)	776	0.0(0)	8.0	775	1030.2	775	1030.2	0.0	10.7	1030.3	53	39	7	218	0	—
	776	0.0(0)	8.0													
	853	0.0(0)	0.0													
20(2)h	701	22.4(1)	22.0	701	196.5	701	196.5	−0.9	0.2	196.6	22	72	5	156	0	—
	777	0.8(1)	16.0													
	777	15.4(1)	0.0													
20(3)	802	3.7(3)	10.7	—	—	—	—	—	—	‡	81	16	13	4	1	155
	802	3.7(3)	10.7													
	802	3.7(3)	10.7													
10(1)	700	22.8(6)	37.0	1006	894.9	1006	894.9	44.0	44.0	‡	57	41	312	12	236	120
	855	15.6(8)	42.0													
	851	24.9(5)	36.0													
10(2)	851	20.6(6)	16.0	699	1063.2	699	1063.2	−0.7	−0.7	1063.3	48	48	8	312	0	—
	855	12.6(6)	31.0													
	853	17.8(6)	0.0													
10(2)h	701	29.2(6)	21.0	779	338.7	779	338.7	0.3	10.8	338.8	39	51	35	100	23	—
	701	29.2(6)	21.0													
	701	29.2(6)	21.0													
10(3)	802	14.5(7)	9.0	749	3133.7	749	3133.7	0.0	6.8	3133.8	65	30	11	134	0	—
	851	8.7(3)	11.7													
	854	15.8(8)	0.0													

is both the average difference and (in parentheses) the number of times the violation took place.

—lowest deviation from the imposed $\mathcal{HP}$ equity constraints, referred to as violation (2). Reported is the amount of deviation from the constraint. For example, under $\delta_{\mathcal{HP}} = 20$ minutes for 1 day, the first heuristic solution reports a violation of 16 minutes. Therefore the maximum difference in delivery time between members of $\mathcal{HP}$ is $16 + 2\delta_{\mathcal{HP}} = 56$ minutes.

For each solution, we report the routing cost and corresponding equity violations, with ties broken in consideration of the above ordering.

- (Grouped Column 3) First Integer Solution. This group of columns presents information about the first feasible integer solution (excluding heuristic solutions) identified by our procedure. Included in these results is the objective function value (given on a per day basis) and the CPU requirement to obtain the solution.
- (Grouped Column 4) Best Integer Solution. Following the format for the first solution, this column presents the corresponding information for the best integer solution identified. Additionally, the best solution's deviation from the linear programming bound, both with and without a cut on the number of vehicles is shown.
- (Grouped Column 5) CPU Requirement. Included is the total CPU requirement with percentage requirements for both the revised, bounded simplex and column generating components of our procedure. The small unreported percentage represents time taken for managing the b&b as well as input-output. Problems reaching a CPU limit have the symbol ‡. When searching for equity over one day the CPU limit

is 4,000 seconds for the 25 and 35 aggregate customer problems. For multiple day scenarios and 50 aggregate customer problems the limit is 8,400 seconds.

- (Grouped Column 6) Branch & Bound. Reported is the total number of nodes explored as well as the number fathomed by bound and infeasibility. When reaching the CPU limit, we report the number of nodes in the b&b tree that remain active at the time of termination.

We now separately discuss our empirical findings for 25, 35, and 50 aggregate customer data sets.

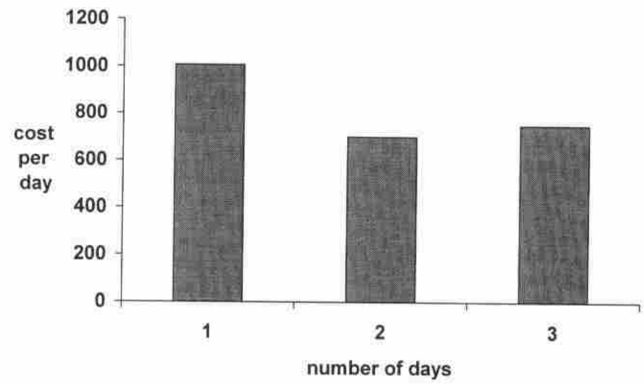

Fig. 4. The cost of equity over multiple days.

4.1. 25 Aggregate Customers

Table I shows the empirical results for 25(1). The first row reports the solution for an equity level of $\delta_{\mathcal{HCP}} = 30$ under 1 day (the *lp* relaxation identifies the optimal integer solution). In the second row the level of equity is tighter and the procedure requires the b&b procedure to find an integer solution. After obtaining the integer solution, all remaining active nodes fathom by bound. This is typical when the number of vehicles from the *lp* relaxation is the best number of vehicles. If the rounded number of vehicles is even one unit away from the required number, the b&b must explore an excessive number of solutions before concluding it has the best solution. Such is the case for the row labeled 10(1), $\delta_{\mathcal{HCP}} = 10$ for 1 day. In this situation, integer solutions require a number of vehicles that is more than the *lp* would commit.

As for the *lp* relaxation to our master problem ($\mathcal{CSP}$), the number of vehicles and objective function value remains the same regardless of the imposed equity level. This holds true in general since the *lp*, for all practical purposes, can assume an unlimited number of days.

For problem 25(1)–10(1), the best solution is sought given any available cost. For the remaining problems, within the b&b, we stipulate a bound on our willingness to pay for equity. This bound is an additional 10% of the minimum determined operating cost. Although high cost integer solutions may not be identified, it is unlikely that these solutions would be implemented.

For this problem (see Figure 4), a tight level of equity can be easily enforced under a multiple day routing scenario. Figure 5 shows the significant increase in computation time for $\delta_{\mathcal{HCP}} = 20$ which is commonly required for multiple days. Fortunately, the multiple day heuristic performs well in limited time. In fact, computation times for the two day heuristic (which include just the additional time for running the second day) often runs more effectively than an ordinary one-day problem. This result can be partially explained by the effect of the fixed routes for one-day with imposed equity making a number of second-day routes impossible. Furthermore, the $\mathcal{CSP}$ *lp* relaxation appears to supply the subproblem with better estimates of favorable nodal arrival costs when compared to solutions where the equity is not as restrictive.

Within this table, the negative numbers in the column measuring the deviation from the *lp* bounds with and without the cut indicates our procedure did not obtain the optimal *lp* relaxation to $\mathcal{CSP}$ prior to branching (*i.e.*, the $\mathcal{SPT}$ heuristic did not identify the most favorable reduced cost).

Tables II and III respectively show the results obtained for 25(2) and 25(3). Again, the heuristic solution performs well for tight levels of equity and the 1 day tight equity problems and multiple day problems require significant computation time.

Table II shows 20(2)h and 10(2)h obtain a solution with the same cost as 20(2) and 10(2) respectively. Although they obtain the same cost solution, the percent deviation from the *lp* bound is significantly

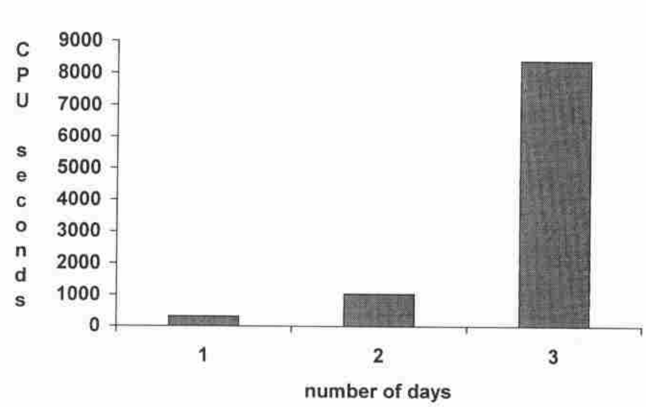

Fig. 5. Computational effort for multiple days.

TABLE II

Empirical Results for 25(2)

	Heuristic Solutions			First Int. Soln.		Best Integer Solution				CPU Required			Branch & Bound			
		Equ. viol.						% deviation								
Equity Level	Obj. val.	(1)	(2)	Obj. val.	CPU (secs)	Obj. val.	CPU (secs)	cut	w/o cut	Total	% simp.	% gen. col.	No. expl.	No. fath. (bound)	No. fath. (infea.)	No. act.
30(1)	692	0.0(0)	0.0	692	10.6	692	10.6	0.1	13.0	266.4	36	55	—	—	—	—
	692	0.0(0)	0.0													
	692	0.0(0)	0.0													
20(1)	692	12.3(3)	10.0	695	463.9	695	463.9	0.6	12.0	463.9	34	59	11	152	2	—
	692	4.5(3)	11.0													
	695	8.6(4)	7.0													
20(2)	691	5.1(3)	10.0	692	2194.8	692	2194.8	7.0	12.0	‡	69	28	37	337	2	267
	695	2.6(3)	0.0													
	695	2.6(3)	0.0													
20(2)h	692	16.1(5)	9.0	693	10.3	693	10.3	0.3	0.3	378.1	26	68	—	—	—	—
	693	11.2(3)	0.0													
	693	11.2(3)	0.0													
20(3)	692	5.2(2)	10.7	691	3913.6	691	3913.6	8.0	14.0	‡	78	19	30	174	1	680
	695	0.0(0)	0.0													
	695	0.0(0)	0.0													
10(1)	692	14.2(6)	39.0	999	534.8	848	640.5	23.0	23.0	‡	86	10	401	80	247	151
	692	14.2(6)	39.0													
	694	21.9(5)	31.0													
10(2)	692	20.0(6)	13.0	694	2820.0	694	2820.0	13.0	13.0	‡	78	21	29	151	0	299
	692	9.7(6)	11.5													
	848	13.9(5)	0.0													
10(2)h	692	19.5(4)	40.0	771	373.6	694	920.5	0.5	0.5	920.6	66	29	206	200	168	—
	692	19.5(4)	40.0													
	694	22.2(7)	24.0													
10(3)	691	8.4(6)	30.3	742	5450.9	691	8253.6	8.0	13.0	‡	73	25	40	270	10	447
	691	8.4(6)	30.0													
	694	14.6(7)	16.0													

different. For 20(2)h and 10(2)h, the first day's solution is fixed and we use the *lp* bound found under this restriction.

For these 25 customer sets, a number of non-reported statistics are also of interest. Of these, the number of iterations required between $\mathscr{CSP}$ and the route generating heuristic is typically about 20, increasing only slightly for more days and tighter equity. The solution to these 25 customer problems typically requires about 200 seconds to reach the *lp* bound. When considering multiple days, although the number of calls to our route generating subproblem increases only modestly, the required time to reach the *lp* bound increases significantly to the approximate levels of 2,000 and 3,000 seconds for the 2 and 3 day problems respectively.

From the computational results of 25(1), 25(2), and 25(3), we conclude that our heuristic procedure for multiple day routing performs well while not requiring excessive computation time. Since our original algorithm run over multiple days requires significant computation, for the remaining problems, we only report the heuristic performance for tight levels of equity over 2 days. Furthermore, it can easily be concluded from these 25 node sets that the equity level of 10 over 1 day is unreasonable for a moderate increase in cost. We therefore only consider this level of equity under multiple day routing.

4.2. 35 Aggregate Customers

Following the format of the 25 node problems, Tables IV through VI present computational experience with 35(1) through 35(3). As stated earlier, the CPU limitation is 8,400 seconds. For the one day

TABLE III

Empirical Results for 25(3)

	Heuristic Solutions			First Int. Soln.		Best Integer Solution				CPU Required			Branch & Bound			
Equity Level	Obj. val.	Equ. viol. (1)	Equ. viol. (2)	Obj. val.	CPU (secs)	Obj. val.	CPU (secs)	% deviation cut	% deviation w/o cut	Total	% simp.	% gen. col.	No. expl.	No. fath. (bound)	No. fath. (infea.)	No. act.
30(1)	688	3.0(2)	0.0	691	10.0	691	10.0	0.4	15.0	459.2	30	62	—	—	—	—
	690	0.0(0)	0.0													
	688	3.0(2)	0.0													
20(1)	690	16.5(4)	24.0	692	1401.5	692	1401.5	0.4	16.0	1401.6	59	37	82	298	20	—
	846	7.5(6)	19.0													
	691	20.8(3)	10.0													
20(2)	690	9.5(1)	1.0	690	2337.8	690	2337.8	12.5	14.3	‡	72	24	33	28	0	225
	696	0.0(0)	0.0													
	990	9.5(1)	0.0													
20(2)h	691	15.2(5)	21.0	692	389.7	692	389.7	0.5	14.9	389.8	26	66	8	227	2	—
	692	11.2(2)	7.0													
	694	16.0(5)	3.0													
20(3)	691	5.2(2)	30.3	692	4986.3	692	4986.3	8.2	13.6	‡	61	36	11	135	0	250
	796	0.0(0)	0.0													
	796	0.0(0)	0.0													
10(1)	690	19.6(7)	42.0	—	—	—	—	—	—	‡	69	27	257	0	212	135
	690	13.6(5)	32.0													
	690	13.6(5)	32.0													
10(2)	690	10.9(5)	36.5	692	3959.7	692	3959.7	12.8	14.3	‡	60	35	34	233	10	304
	694	14.2(7)	33.5													
	772	20.3(5)	20.5													
10(2)h	689	35.7(5)	48.0	691	630.4	691	630.4	0.4	14.8	630.8	19	75	3	221	0	—
	692	41.8(3)	21.0													
	691	19.8(6)	35.0													
10(3)	690	19.6(6)	29.0	691	7457.8	691	7457.8	8.2	14.2	‡	76	22	18	66	2	548
	691	10.4(5)	30.7													
	690	11.0(8)	6.7													

TABLE IV

Empirical Results for 35(1)

	Heuristic Solutions			First Int. Soln.		Best Integer Solution				CPU Required			Branch & Bound			
Equity Level	Obj. val.	Equ. viol. (1)	Equ. viol. (2)	Obj. val.	CPU (secs)	Obj. val.	CPU (secs)	% deviation cut	% deviation w/o cut	Total	% simp.	% gen. col.	No. expl.	No. fath. (bound)	No. fath. (infea.)	No. act.
30(1)	1032	2.4(1)	0.0	1032	26.1	1032	26.1	0.0	15.0	1443.4	32	64	—	—	—	—
	1042	1.4(2)	1.0													
	1032	2.4(1)	0.0													
20(1)	1032	9.6(6)	17.0	1035	3521.8	1035	3521.8	18.0	18.0	‡	27	70	9	194	2	63
	1032	6.7(6)	18.0													
	1034	17.0(4)	14.0													
20(2)h	1034	49.6(5)	47.0	1036	56.7	1036	56.7	0.2	15.7	2294.3	20	74	6	219	0	—
	1117	8.9(1)	23.0													
	1117	8.9(1)	23.0													
10(2)h	1033	41.2(8)	64.0	1114	3302.4	1038	3848.0	0.5	17.3	3848.2	45	51	85	216	47	—
	1035	20.3(8)	69.0													
	1036	26.6(9)	33.0													

TABLE V

Empirical Results for 35(2)

Equity Level	Heuristic Solutions Obj. val.	Heuristic Solutions Equ. viol. (1)	Heuristic Solutions Equ. viol. (2)	First Int. Soln. Obj. val.	First Int. Soln. CPU (secs)	Best Integer Solution Obj. val.	Best Integer Solution CPU (secs)	% deviation cut	% deviation w/o cut	CPU Required Total	CPU Required % simp.	CPU Required % gen. col.	Branch & Bound No. expl.	Branch & Bound No. fath. (bound)	Branch & Bound No. fath. (infea.)	Branch & Bound No. act.
30(1)	871	3.0(2)	6.0	872	2170.1	872	2170.1	0.1	4.0	2170.2	28	66	10	379	4	—
	1032	0.7(1)	0.0													
	1031	1.9(1)	0.0													
20(1)	873	97.2(7)	28.0	—	—	—	—	—	—	‡	53	44	23	2	15	245
	873	7.3(7)	28.0													
	1031	9.8(6)	21.0													
20(2)h	873	11.6(3)	25.0	950	2525.5	875	4892.4	0.3	3.4	4892.5	56	39	39	337	10	—
	954	6.2(1)	2.0													
	953	7.8(2)	0.0													
10(2)h	872	27.4(7)	44.0	—	—	—	—	—	—	‡	66	32	370	0	312	295
	953	14.0(8)	14.0													
	953	17.2(7)	11.0													

TABLE VI

Empirical Results for 35(3)

Equity Level	Heuristic Solutions Obj. val.	Heuristic Solutions Equ. viol. (1)	Heuristic Solutions Equ. viol. (2)	First Int. Soln. Obj. val.	First Int. Soln. CPU (secs)	Best Integer Solution Obj. val.	Best Integer Solution CPU (secs)	% deviation cut	% deviation w/o cut	CPU Required Total	CPU Required % simp.	CPU Required % gen. col.	Branch & Bound No. expl.	Branch & Bound No. fath. (bound)	Branch & Bound No. fath. (infea.)	Branch & Bound No. act.
30(1)	869	5.8(2)	5.0	870	2509.5	870	2509.5	0.1	7.3	2059.6	26	68	9	378	3	—
	874	0.1(1)	0.0													
	874	0.1(1)	0.0													
20(1)	869	5.5(7)	24.0	—	—	—	—	—	—	‡	33	63	20	1	10	208
	869	5.5(7)	24.0													
	873	15.6(4)	17.0													
20(2)h	870	8.8(3)	18.0	876	2009.9	876	2009.9	0.4	7.6	2010.0	18	76	5	380	0	—
	873	0.0(0)	0.0													
	873	0.0(0)	0.0													
10(2)h	870	8.6(7)	48.0	1030	4306.6	875	6228.8	3.5	7.5	6228.9	42	52	82	254	48	—
	870	8.6(7)	48.0													
	951	16.1(7)	38.0													

TABLE VII

Empirical Results for 50(1)

Equity Level	Heuristic Solutions Obj. val.	Heuristic Solutions Equ. viol. (1)	Heuristic Solutions Equ. viol. (2)	First Int. Soln. Obj. val.	First Int. Soln. CPU (secs)	Best Integer Solution Obj. val.	Best Integer Solution CPU (secs)	% deviation cut	% deviation w/o cut	CPU Required Total	CPU Required % simp.	CPU Required % gen. col.	Branch & Bound No. expl.	Branch & Bound No. fath. (bound)	Branch & Bound No. fath. (infea.)	Branch & Bound No. act.
30(1)	1382	5.6(1)	4.0	1387	91.4	1387	91.4	0.3	12.0	4183.5	37	58	—	—	—	—
	1397	2.0(2)	4.0													
	1397	2.2(1)	0.0													
20(2)h	1385	9.0(6)	22.0	1383	4214.8	1383	4212.8	0.1	12.0	4213.0	36	58	12	473	3	—
	1388	4.6(1)	0.0													
	1388	4.6(1)	0.0													

TABLE VIII
Empirical Results for 50(2)

	Heuristic Solutions			First Int. Soln.		Best Integer Solution				CPU Required			Branch & Bound			
		Equ. viol.						% deviation								
Equity Level	Obj. val.	(1)	(2)	Obj. val.	CPU (secs)	Obj. val.	CPU (secs)	cut	w/o cut	Total	% simp.	% gen. col.	No. expl.	No. fath. (bound)	No. fath. (infea.)	No. act.
30(1)	1223	4.1(14)	7.0	1379	83.9	1225	7829.9	0.1	3.0	7830.0	48	48	14	536	1	—
	1379	0.0(0)	0.0													
	1379	0.0(0)	0.0													
20(2)h	1225	23.2(9)	31.0	1384	8363.4	1384	8363.4	6.7	10.6	‡	35	61	34	12	19	480
	1305	1.4(1)	0.0													
	1305	1.4(1)	0.0													

TABLE IX
Empirical Results for 50(3)

	Heuristic Solutions			First Int. Soln.		Best Integer Solution				CPU Required			Branch & Bound			
		Equ. viol.						% deviation								
Equity Level	Obj. val.	(1)	(2)	Obj. val.	CPU (secs)	Obj. val.	CPU (secs)	cut	w/o cut	Total	% simp.	% gen. col.	No. expl.	No. fath. (bound)	No. fath. (infea.)	No. act.
30(1)	1384	4.4(3)	3.0	1383	4108.0	1383	4108.0	0.0	12.0	4108.8	33	61	2	63	0	—
	1390	1.0(3)	0.0													
	1389	1.0(3)	0.0													
20(2)h	1383	22.0(8)	36.0	—	—	—	—	0.4	13.3	‡	31	63	26	0	14	350
	1388	0.0(0)	0.0													
	1388	0.0(0)	0.0													

problem with equity 20, all problems fail to identify a solution within an additional 10% of the minimum operating cost in the required time. A solution is found for 35(1) while obtaining the solution to the *lp* relaxation but it is not within the 10% used in the b&b. It appears that this level of equity is not easily established for a single day. Our heuristic again demonstrates that this level of equity is easily handled within the multiple day scenario.

4.3. 50 Aggregate Customers

Tables VII through IX show the results for the 50 customer data sets. Since the computation time increases significantly for equity levels below 30, we provide solutions for the 1 day problem under loose equity and the heuristic solution under the equity level 20. From 50(3), another use for the schedule generating heuristic is apparent, since a feasible schedule is found where the b&b failed to identify a solution within the time limits. Since the *lp* solution had a lower objective function value at the time of generation, the solution would not have been recorded if it had not been for the heuristic.

Since it typically requires over 4,000 seconds to solve the *lp* relaxation, the b&b phase did not have sufficient time to explore many possibilities. Thus, 50(2)–20(2)h and 50(3)–20(2)h fail to terminate before the 8,400 second time limit. Even with the time limit verifiably good solutions are found.

5. SUMMARY

THIS PAPER CONSIDERS a multiple vehicle traveling salesman problem with time windows and equity constraints on customer delivery times, cast over a multiple day framework. It presents a column generation framework where the master problem is a constrained set partitioning problem and the subproblem is a shortest path problem with time windows and an additional cost on node arrival times. The paper provides an optimal column generation based solution procedure along with heuristic variations. It provides a detailed discussion of test problem development—test problems come from an overnight delivery situation, which is the motivation behind this new formulation. Computational results indicate equity between average customer deliver times can be incorporated at little or no extra cost, using a multiple day framework.

ACKNOWLEDGMENTS

THE WORK OF the first author was supported, in part, by separate grants from Federal Express and the Naval Postgraduate School Research Initiation Program; this support is gratefully acknowledged. The second author's work is part of his efforts on initiative #6 "Spatial Decision Support Systems," of the National Center for Geographic Information and Analysis (NCGIA); NCGIA is supported by a grant from the National Science Foundation (grant SES-88-10917)—this support is gratefully acknowledged. Finally, the authors thank an anonymous Associate Editor for his/her useful comments.

REFERENCES

AGARWAL, Y., K. MATHUR, AND H. M. SALKIN, "A Set Partitioning—Based Exact Algorithm for the Vehicle Routing Problem," *Networks* **19,** 731–749 (1989).

BODIN, L. D., B. L. GOLDEN, A. A. ASSAD, AND M. O. BALL, "Routing and Scheduling of Vehicles and Crews: The State of The Art," *Computers and Operations Research* **10,** 62–212 (1983).

DANTZIG, G. B. AND P. WOLFE, "Decomposition Principle for Linear Programs," *Operations Research* **8,** 101–111 (1960).

DANTZIG, G. B. AND P. WOLFE, "The Decomposition Algorithm for Linear Programs," *Econometrica* **29,** 767–778 (1961).

DELL, R. F., "The Development of Equitable Vehicle Routes for Overnight Parcel Deliveries" Ph.D. dissertation, Department of Industrial Engineering, SUNY at Buffalo, 1990.

DESROCHERS, M., J. DESROSIERS, AND M. SOLOMON, "A New Optimization Approach for the Vehicle Routing Problem with Time Window," presented at the ORSA/TIMS Conference, Las Vegas, (1990).

DESROCHERS, M. AND F. SOUMIS, "A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows," *INFOR* **26,** 191–212 (1988).

DESROSIERS, J., F. SOUMIS, AND M. DESROCHERS, "Routing With Time Windows by Column Generation," *Networks* **14,** 545–565 (1984).

GOLDEN, B. L. AND A. A. ASSAD, "Perspectives on Vehicle Routing: Exciting New Developments," *Operations Research* **34,** 803–810 (1987).

GOLDEN, B. L. AND A. A. ASSAD, *Vehicle Routing: Methods and Studies,* North-Holland, New York, 1988.

GOPALAN, R., K. S. KOLLURI, R. BATTA, AND M. H. KARWAN, "Modeling Equity of Risk In the Transportation of Hazardous Material," *Operations Research* **38,** 961–973 (1990).

KEENEY, R. L., "Equity and Public Risk," *Operations Research* **28,** 527–534 (1980a).

KEENEY, R. L., "Utility Functions for Equity and Public Risk," *Management Science* **26,** 345–353 (1980b).

KOSKOSIDIS, Y. A., W. B. POWELL, AND M. M. SOLOMON, "An Optimization Based Heuristic for Vehicle Routing with Time Window Constraints," *Transportation Science* **26,** 69–85 (1992).

KOLESAR, P., W. WALKER, AND J. HAUSNER, "Determining the Relationship Between Fire Engine Travel Times and Travel Distances in New York City," *Operations Research* **23,** 614–627 (1975).

LARSON, R. C., "Perspectives On Queues: Social Justice and the Psychology of Queueing," *Operations Research* **35,** 895–905 (1987).

LAWLER, E. L., J. K. LENSTRA, A. H. G. RINNOOY KAN, AND D. B. SHMOYS, *The Traveling Salesman Problem,* John Wiley, New York, 1985.

MAGNANTI, T., "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects," *Networks* **11,** 179–214 (1981).

MANDELL, M. B., "Modelling Effectiveness Equity Tradeoffs in Public Service Delivery Systems," *Management Science* **37,** 467–482 (1991).

SOLOMON, M. M., "Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms" Ph.D. Dissertation, University of Pennsylvania, 1984.

(Received: December 1992; revisions received: February 1995, April 1995; accepted: April 1995)